# Modélisation par les grammaires de graphes de la génération de la diversité dans les familles de produits.

Blaise Mtopi Fotso, Maryvonne Dulmet, Eric Bonjour

# Modélisation par les grammaires de graphes de la génération de la diversité dans les familles de produits

**Blaise Mtopi Fotso*,** — Maryvonne Dulmet* — Eric Bonjour***

** Département AS2M de FEMTO ST – UFC / ENSMM*
*UMR CNRS 6174*
*24 rue Alain Savary 25000 Besançon - France*
*fmtopi@yahoo.fr, {mdulmet, ebonjour}@ens2m.fr*

*** Laboratoire d'Ingénierie des Systèmes Industriels et de l'Environnement - LISIE*
*Département de Génie Mécanique et Productique (GMP), IUT Fotso Victor de Bandjoun, Université de Dschang*
*B.P. 134 Bandjoun - Cameroun*

RÉSUMÉ. *Cet article présente une méthode de génération de la diversité dans une famille de produit à l'aide de grammaire de graphe à attributs programmée. A partir de la modélisation d'un produit de base, cet outil permet d'obtenir l'architecture de la famille et de générer les variantes en appliquant quatre types d'opérateurs. Nous l'illustrons avec la modélisation d'éléments d'une cuisine intégrée.*

ABSTRACT. *This paper proposes a methodology to generate the diversity in a products family with Programmed Attributed Graph Grammars. Starting from the model of a Base-Product, this tool allows to obtain the family architecture and to generate the variants, using four types of operators. We illustrate our purpose with some parts of a kitchen.*

MOTS-CLÉS : *Famille de Produits, Architecture, Modularité, Diversité, Grammaires de graphe.*

KEYWORDS: *Products Family, Architecture, Modularity, Diversity, Graph Grammars.*

## 1. Introduction

Dans le contexte concurrentiel actuel, les industriels doivent concevoir et réaliser une grande variété de produits pour répondre à des besoins de personnalisation accrue demandés par les clients.

Ils doivent, pour cela, déterminer la diversité commerciale, attendue par les clients, générer des solutions techniques en maîtrisant la diversité technique des produits et concevoir un système industriel suffisamment flexible pour permettre la réalisation d'un ensemble de produits tout en maîtrisant la diversité industrielle (Agard et Tollenaere, 2002 ; Agard et Tollenaere, 2003). Une question difficile concerne alors la manière de produire et de maîtriser ces diversités dans des délais et des coûts acceptables de conception et de réalisation.

Le développement de famille de produits a été reconnu comme un moyen efficace pour réaliser une économie d'échelle, pour accommoder une diversité croissante de produits à travers des marchés spécifiques et pour réduire les risques de développement par la réutilisation des éléments provenant des offres et activités antérieures.

Notre propos dans ce texte est de montrer comment à partir d'un produit conçu modulairement, l'architecture du produit et celle de sa famille peuvent être représentées par des grammaires de graphes à attributs programmés , puis générer la diversité des produits de la famille.

La section 2 présente une synthèse sur les familles de produits et les architectures de familles de produit, comme réponse à la demande de variété des clients.

La section 3 s'intéresse à la modélisation des familles de produits par les graphes. En effet, les graphes permettent de modéliser une grande variété d'objets et de situations en raison de la correspondance entre le langage formel et la famille de produits. Un bref rappel est fait sur les grammaires formelles et les grammaires de web.

La section 4 présente la modélisation par les grammaires de graphes à attributs programmés d'une architecture d'une famille de produits, ainsi que la génération des variantes. Ceci est illustré par un exemple industriel. Il s'agit de meubles de cuisine intégrée qui est par excellence un produit à forte diversité et à caractère modulaire.

## 2. Les familles de produits

Les problématiques concernant la conception d'une famille de produits ont été abordées par différents auteurs et selon divers points de vue : positionnement marketing, conception d'une plateforme produit, conception du système industriel et conception de la chaîne logistique (Jiao *et al.,* 2006 ; Lamothe *et al.,* 2006). De ce fait, le concept de famille de produits a été défini de façon variée. Salvolaine *et al.* (1995) affirment ainsi que l'une des difficultés de l'étude du concept de famille de

produits réside dans l'absence d'une terminologie *universelle*. Les termes les plus souvent utilisés comme *famille de produits* et *variante de produits* ont des significations parfois différentes.

Du point de vue positionnement marketing, des auteurs comme (Rampersad, 1994 ; Ulrich et Tung, 1991 ; McKay *et al,* 1996) considèrent une famille de produits comme un ensemble qui identifie les différences et les similarités entre les produits individuels formant la gamme des produits.

L'aspect plateforme produit permet de considérer une famille de produits comme un ensemble de produits manufacturiers à l'intérieur d'une même entreprise et pour lesquelles les fonctions principales sont identiques (Dufrène, 1991 ; Perrard, 1993).

L'approche relative au système de fabrication permet à Miller et Liberatore (1993), Laokko et Mäntylä (1994), Fan et Liu (1999) de définir les structures de famille de produits sur la base de processus de routage, de fabrication et/ou d'assemblage similaires. Ainsi les familles de produits représentent des *packages* intermédiaires de produits finaux partageant des caractéristiques identiques en termes de processus de production.

Pour De Lit (2001), un autre terme utilisé est *groupe de produits* qui représente un nombre de produits ayant une ou plusieurs caractéristiques communes permettant une combinaison du processus de planification et de contrôle. Toutefois, un *groupe de produits* est semblable à une *famille de produits*, mais les industriels et universitaires utilisent le second terme plutôt que le premier.

Plus généralement, certains auteurs, (Meunier, 1989), (Meyer, 1997 ; Meyer et Lopez, 1995 ; Erens, 1996 ; Erens *et al.*, 1997) les définissent comme un ensemble de produits possédant des technologies communes qui concourent directement à un ensemble d'applications commerciales. Ils concluent qu'une famille de produits est un concept crée pour le marché, répondant à l'ensemble des attentes des consommateurs par l'introduction de la variété dans la définition de l'architecture du produit et dans celle des processus de fabrication.

Ces définitions mettent l'accent sur les processus communs, les caractéristiques communes et les fonctions communes ou encore le partage de technologies communes.

La définition la plus communément admise pour une famille de produit se réfère à un ensemble de produits semblables qui dérivent d'une plateforme commune et qui ont des fonctionnalités spécifiques répondant aux besoins des clients (Meyer, 1997).

Un état de l'art complet se trouve dans (Agard, 2004).

### *2.1 Conception de plate-forme et familles de produits*

Pour concevoir une famille de produits deux caractéristiques importantes sont à prendre en compte la modularité (Martin et Ishii, 1996 ; Ulrich et Tung 1991) et la standardisation (Ulrich, 1995).

Une famille de produits permet de produire des variantes utilisant une combinaison de différents modules. Ce sont à la fois la modularité et la

standardisation qui permettent de concevoir une grande variété de produits à partir des mêmes modules, (commonalité et réutilisation).

Chaque variante de la famille est développée pour répondre à un besoin sur un segment de marché et les variantes partagent une structure commune qui constitue la plate - forme de la famille de produits (Erens et al, 1997).

La signification du concept de plate-forme varie en fonction des besoins Certaines définitions se focalisent principalement sur le produit (ou artefact), alors que d'autres explorent le concept de plate-forme en termes de chaînes de valeur des entreprises (Dooley et O'Sullivan, 2000).

Selon les approches, une plate-forme est considérée comme le support du développement d'une famille de produits ou encore comme une collection d'éléments partagés par plusieurs produits semblables (Mc Graphs 1995 ; José 2005). En conséquence, le problème est d'identifier les éléments communs pour une famille de produits. Il s'agit de réaliser l'extraction de ces éléments des produits communs, des caractéristiques et/ou sous systèmes qui sont stables et bien maîtrisés. Siddique et al. (1999) ont utilisé les grammaires de graphes pour identifier la plate-forme commune d'un ensemble de produits similaires et spécifier le portefeuille de produits supporté par la plate-forme.

L'approche plate-forme est une solution adoptée pour répondre au dilemme standardisation / différenciation. Elle repose sur une architecture modulaire des produits, la plate-forme étant une forme avancée de gestion de cette modularité.

### *2.2 Conception modulaire de produits*

Qu'il fasse partie d'une famille ou non (produit unique), un produit complexe peut être conçu de façon modulaire.

La modularité est un concept de structuration permettant de gérer cette complexité. Une étude de la littérature montre qu'elle se décline sous deux formes.

La première s'articule autour de la décomposition d'un système en plusieurs sous-systèmes dont le but est de faciliter les activités de conception, de production et d'usage. Pour Baldwin et Clark (1997), elle permet le développement d'un produit complexe à partir de sous-systèmes qui doivent fonctionner comme un ensemble.

La seconde est relative au découplage entre les parties. Ici, il s'agit d'une approche de conception via la standardisation des interfaces des composants d'un produit. Sako (2003) illustre cela clairement en identifiant la modularité par les interfaces standardisées et ouvertes qui permettent d'augmenter la possibilité de *mix and match*[1] .

[1] Le *mix and match* est une stratégie permettant de créer une grande variété de produits par la standardisation des composants qui sont destinés à des emplois multiples. L'idée est d'utiliser un minimum de composants pour réaliser un maximum de produits différents.

En conception modulaire, un module pourra être un composant (souvent considéré comme un ensemble de pièces élémentaires) ou un ensemble de composants. Le produit (en entier) pourra être vu aussi comme un module. Deux éléments apparaissent comme fondamentaux dans la conception du produit : l'allocation composants / fonctions et les caractéristiques des interfaces.

La modularité en conception concerne une architecture de produits pour lesquelles les interfaces sont standards. Pour mieux représenter et gérer la diversité, de nombreux auteurs introduisent deux types de modules : les modules communs et les modules distinctifs, ces derniers étant organisés en modules composés et modules primitifs.

### *2.3 Architecture du produit et de la famille de produits*

Nous plaçons le concept de modularité au centre de l'élaboration de l'architecture du produit (Ulrich,1995). Ulrich et Eppinger (1995) affirment que l'architecture du produit est un élément fondamental de la performance d'une entreprise. Ils la définissent comme la façon d'organiser les éléments fonctionnels d'un produit en sous-systèmes et de spécifier les interfaces entre ces sous-systèmes. De ce fait définir l'architecture du produit revient à allouer des fonctions à des composants physiques en isolant trois étapes : la structuration des éléments fonctionnels, l'allocation de ces éléments sur des composants physiques, la spécification des interfaces entre ces composants qui interagissent.
Ils distinguent deux types d'architectures :
- l'architecture modulaire : une architecture est dite modulaire si on a un « *one to one mapping* » entre fonctions et modules (c'est-à-dire lorsqu'à une fonction ne correspond qu'un seul constituant physique) et si les interfaces sont découplées dans le sens où modifier un constituant ne demande pas de reconcevoir les autres interfaces**.** Il en résulte un découplage total entre les constituants.

- l'architecture intégrale (qu'on qualifiera aussi d'intégrative ou d'intégratrice) est une architecture où les constituants sont fortement couplés de telle sorte qu'un changement apporté sur un constituant nécessite l'apport d'adaptation sur d'autres constituants.

Chen et Liu [2005] précisent que l'allocation « *one to one* » des fonctions vers les composants est dépendante du niveau de décomposition réalisé tant dans l'arborescence des fonctions que dans l'arborescence du produit. Ils proposent alors de réaliser l'allocation d'un sous-ensemble de fonctions vers un sous-ensemble de composants.

De nombreuses études ont démontré l'intérêt de l'architecture modulaire pour prendre en compte la diversité et le besoin de flexibilité dans la réalisation des produits (Jiao *et al.,* 2000). L'architecture d'une famille de produits doit définir à la fois ce qui est commun et ce qui est distinctif parmi les membres de la famille, aussi

bien que les mécanismes par lesquels les variantes du produit peuvent être obtenues. De ce fait, l'architecture d'une famille de produits représente la structure conceptuelle et l'organisation logique de la famille selon le double point de vue du consommateur et du concepteur (Zha et Sriram, 2006).

La structure générique d'un produit (SGP) d'une famille de produits fait référence à l'organisation générique de tous les modules qui peuvent se trouver dans la famille (Jiao *et al.,* 1999).

Une architecture de famille de produits (AFP), quant à elle, est matérialisée par les produits de base, les modules distinctifs et les règles de combinaisons qui conduisent aux variantes de produits :

- *Le produit de base (PB)* matérialise les éléments partagés d'une famille de produits. Sur le plan marketing, ces éléments s'instancient aux caractéristiques fonctionnelles communes, tandis que du point de vue technique, ce sont les modules communs matérialisant les structures communes du produit. Il rassemble tous les modules implémentant les fonctions principales de la famille et est considéré comme un « module composé ».

- *Les modules distinctifs (MDs)* sont ceux qui rendent les produits différents les uns des autres. Ils sont à l'origine de la variété pour une famille de produits. Vu du côté technique, ces modules distinctifs peuvent être exprimés en relations structurelles distinctes ayant des performances fonctionnelles différentes. Ce sont des modules composés et modules primitifs. Sur le plan marketing, ils deviendront des caractéristiques fonctionnelles ou des caractéristiques de valeurs pouvant être sélectionnées.
- *Les règles de combinaisons (RCs)* définissent les règles et méthodes de dérivation des variantes du produit.

### *2.4 Gestion de la diversité*

A partir d'une architecture générique de la famille de produits, il devient possible d'obtenir l'ensemble des variantes de la famille à travers le choix des modules distinctifs et l'application des règles, l'objectif étant de trouver un compromis entre la variété et la standardisation, tout en respectant un certain nombre de contraintes.

La définition de "variante" change selon les auteurs. En effet, elle est une conséquence directe de la définition de famille de produits.

Pour Erens, (1996), un "produit variant" est une occurrence, dérivée d'une famille de produits pour répondre à la demande du consommateur. Pour (McKay et al., 1996), une variante est un produit séparé (individuel) conforme à cette famille de produits.

Pour la réalisation de la variété, plusieurs auteurs (Fujita et al., 1999 ; Simpson et al., 1999 ; Ulrich, 1995, Jiao et al., 2004) ont utilisé certaines méthodes permettant

- le choix des composants réalisant les fonctions,
- l'ajustement de plate-forme commune du produit,
- la combinaison des composants.

Deux thèmes s'intéressent à la réalisation de la variété. L'un, se focalise sur les caractéristiques intrinsèques dans la sélection des composants ; l'autre est lié à la méthode de la génération de la variété.

Deux groupes de travaux semblent plus particulièrement découler du premier thème.

Le Product Data Management Group (Männistö et al., 1999) a proposé une structure de produit générique à laquelle s'ajoute un modèle de configuration. Elle se compose d'un modèle explicite contenant les composants, la hiérarchie, les options et variantes et d'un modèle implicite contenant les connaissances sur les compatibilités entre composants et contraintes. Le modèle de configuration contient les exigences du client. Ces auteurs se sont appuyés sur un standard (ISO 10303) pour traiter la représentation et l'échange de modèles de produit (Männistö et al., 1998).

L'équipe du centre de Génie Industriel de l'Ecole d'Albi-Carmaux (Aldanondo et al., 2006) travaille sur la configuration des produits et de son mode opératoire et le résout comme un problème de satisfaction de contraintes. Ensuite elle couple la configuration du produit avec la configuration de sa gamme d'assemblage (Aldanondo et al., 2007).

L'aspect génération de la diversité s'appuie le plus souvent sur l'utilisation de langages formels pour modéliser et générer la diversité. Dans la section 3 nous rappelons comment on peut utiliser des grammaires pour modéliser une famille de produits. Dans la section 4 nous présentons un type de langage formel, les grammaires de graphes et nous l'appliquons à un produit modulaire, puis nous générons l'architecture et les variantes.

### *2.5 Synthèse et positionnement*

Dans cette partie, nous avons analysé les principaux concepts de l'architecture modulaire puis nous avons montré quelques principes d'architecture d'une famille de produits permettant de modéliser et gérer la diversité. De notre étude, il découle qu'un certain nombre de définitions ne sont pas tout à fait stabilisées tant au niveau national, qu'international. Chaque produit individuel dans une famille de produits, i.e. un membre de la famille, est appelé variante du produit. Toutes les variantes du produit partagent des structures et/ou technologies communes qui forment la plate-forme de la famille de produits (point de vue technique).

Pour notre part, nous nous intéressons aux familles de produits, plus particulièrement à la modélisation de leur architecture et à la génération de la variété. Nous retiendrons certaines des définitions présentées auparavant (produit de base, modules distinctifs…) comme base dans la modélisation de l'architecture d'une famille de produits et de la génération de la diversité par les grammaires de graphe.

## 3. Modélisation de l'architecture d'une famille de produits par les graphes

### *3.1. Langage formel et famille de produits*

La figure 1 ci-après proposée par (Djemel, 1994) illustre le parallèle entre la génération des phrases à partir des symboles (mots) en respectant une sémantique liée à une langue donnée et la génération des fonctions d'un produit en considérant les fonctionnalités définies par la conception.

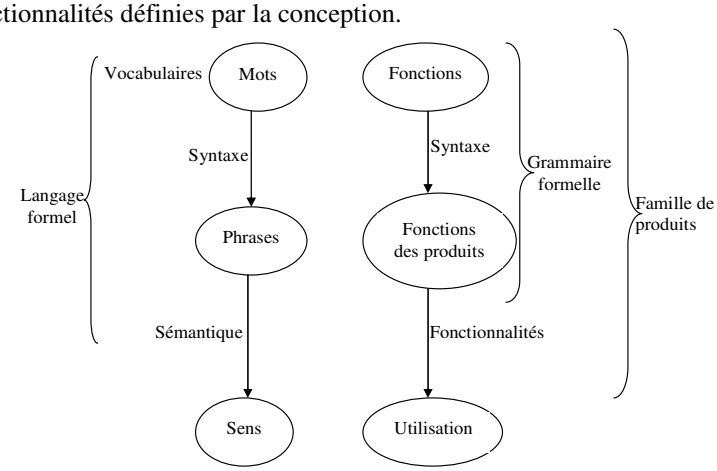



**Figure 1.** *Correspondance entre langage formel et une famille de produits*

Cette figure montre qu'une grammaire définit les éléments et les structures (syntaxes) d'une langue. La grammaire fournit une structure sur laquelle peut être développé un langage. Les interprétations de ce langage sont basées sur une information syntaxique obtenue par l'analyse grammaticale de l'information lexicale inhérente aux composants individuels. Les grammaires peuvent être utilisées pour générer des formes de phrases (des chaînes, des graphiques ou des types de formes variées…) ou pour analyser des phrases dans le but de vérifier la syntaxe pour l'inclure dans le langage.
Cette correspondance montre que les graphes associés à des grammaires peuvent être utilisés pour représenter la structure d'une famille de produits.

### 3.2. *Grammaire formelle et description fonctionnelle*

La modélisation d'un produit développée par A. Bourjault (Bourjault, 1984) est l'une des plus anciennes et est intimement liée à la *modélisation fonctionnelle*. A tout produit, on peut associer un graphe non orienté tel que l'ensemble de ses sommets corresponde de façon biunivoque à l'ensemble des composants élémentaires du produit et l'ensemble de ses arêtes correspond à l'ensemble des liaisons fonctionnelles du produit. Le graphe des liaisons fonctionnelles est un graphe simple où il y a au plus une liaison fonctionnelle entre deux composants élémentaires $C_i$ et $C_j$ et aucune liaison entre un composant $C_i$ et lui-même.

Cette modélisation ne représente que certains aspects du produit, notamment ceux qui sont relatifs à sa structure. Henrioud (1989) a développé la *modélisation opérationnelle* qui est une extension de la modélisation fonctionnelle. Son approche est qualifiée d'approche par constituant et le modèle développé par cette approche est le modèle opératoire. Les chercheurs de CSDL (Charles Stark Draper Laboratory) ont également développé une variante de la modélisation fonctionnelle en s'appuyant sur une modélisation beaucoup plus détaillée (Whitney, 1986).

Dans sa thèse, Dufrène (1991) propose un graphe générique étiqueté s'inspirant du graphe des liaisons fonctionnelles utilisé pour représenter individuellement les produits composés. Il modélise un ensemble de produits qui présentent des différences au niveau de leurs composants et de leurs structures (liaison entre composants). Ces différences sont indiquées par des étiquettes sur un graphe générique. Celui-ci est un graphe associé à une famille de produits ayant les spécificités suivantes :

- les sommets représentent les composants génériques et les composants spécifiques définis sur l'ensemble des composants de la famille,
- il existe une arête entre deux sommets s'il existe au moins un produit de la famille pour lequel les deux sommets sont reliés par une liaison fonctionnelle,
- une étiquette est attachée à chaque sommet et à chaque arête.

La notion de composant générique a la particularité d'être limitée aux seuls composants élémentaires (non décomposables ou primitifs).

Par ailleurs, l'étiquette d'un sommet est un vecteur, de dimension égale au nombre de types de produits de la famille, dans laquelle la i-ème composante vaut 1 si le constituant associé au sommet existe dans le i-ème type de produit et vaut 0 dans le cas contraire. Cette définition suppose que les types de produits de la famille soient rangés de manière à avoir les composants de même rang dans les différentes étiquettes et correspondant toujours au même type de produit.

De même, l'étiquette d'une arête est un vecteur dont la dimension est le nombre de type de produits de la famille, la i-ème composante valant 1 si la liaison fonctionnelle modélisée existe dans le i-ème type de produit et valant 0 dans le cas contraire. L'ordre des types de produits choisis pour déterminer les étiquettes des sommets doit également être utilisé pour celles des arêtes.

L'intérêt présenté par les grammaires formelles est qu'elles permettent une représentation graphique des règles de décomposition et de substitution qui décrivent

la famille. Cependant, la difficulté réside dans la gestion des problèmes de mise à jour du modèle d'une famille de produits lors de l'utilisation du graphe générique étiqueté

### 3.3. *Grammaires de webs*

Pfalz et Rosenfeld (1969) ont introduit la *grammaire de web* pour laquelle le langage est un ensemble de graphes étiquetés (les *webs*) et les combinaisons possibles sont remplacées par les règles de productions. Tout graphe pouvant être généré par application des productions sur un graphe de départ constitue le langage de cette grammaire de graphe. Dans une telle représentation, le modèle de famille est décrit par une grammaire de web et le langage généré par cette grammaire est l'ensemble de types de produits de la famille. Rosenfeld et Milgram (1972) ont introduit les notions de graphes orientés et de graphes non orientés.

Les principes généraux qui sous tendent les grammaires de web sont énoncés ci-après.

Une grammaire de web est une grammaire de graphes étiquetés (c'est-à-dire que les nœuds possèdent des identificateurs) définie par le quadruplet suivant :

$$G_W = (V, V_T, S, P) \qquad [1]$$

Où

$V$ est un ensemble fini non vide, appelé vocabulaire

$V_T$ est un sous-ensemble non vide de, appelé vocabulaire terminal

$V_N = V - V_T$

$S \in V_N$ est le symbole initial

$P$ est un ensemble fini non vide de triplets $(\alpha, \beta, f)$ qui sont appelés règles de réécriture, avec

$$\begin{cases} \alpha \text{ et } \beta \text{ sont des subwebs de } V \\ f : N_\beta \times N_\alpha \rightarrow 2^V \end{cases} \qquad [2]$$

Les *subwebs* sont des sous-ensembles des graphes étiquetés (les *webs*). La production de *web* consiste à remplacer un *subweb* par un autre *subweb*.

$f$ est une application de $N_\beta \times N_\alpha$ dans $2^v$ (l'ensemble des sous-ensembles de V).

$N_\beta, N_\alpha$ sont des ensembles finis non vides contenant les nœuds de $\beta$ *et* $\alpha$ respectivement.

Une règle de réécriture d'une grammaire de Web $(\alpha, \beta, f)$ sera qualifiée de *contextuelle* s'il existe un élément non terminal $A$ du Web $\alpha$ tel que $\alpha - \{A\}$ est un subweb de $\beta$. Si le Web $\alpha$ n'est constitué que d'un point singulier, la règle $(\alpha, \beta, f)$ est dite *hors - contexte*. La modélisation d'une famille de produits par une

grammaire de web peut se faire par l'application de trois types de règles de production.
La règle de type concerne l'instanciation des composants génériques par des composants spécifiques.
La règle de type 2 sert principalement
- à instancier un nœud par des composants polystructurels
- à ajouter des composants optionnels.
La règle de type 3 sert par exemple à ajouter une liaison partielle entre deux composants.

Nous venons de présenter deux modélisations possibles pour des familles de produits à base de grammaires formelles et de grammaires de webs, dans chacun des cas, les familles peuvent être décrites selon les deux aspects fonctionnel et matériel.

L'avantage de ces grammaires est de fournir un graphe générique d'une famille de produits, ce qui représente une étape importante pour la modélisation.

Pour les grammaires de webs, les contraintes portant sur les combinaisons de constituants génériques sont difficiles à appréhender (Mtopi *et al.,* 2004a). De même, l'association des informations contenues dans un modèle avec celles d'autres modèles (fonctionnel, géométrique, d'assemblage) se fait avec beaucoup de difficultés (Mtopi *et al.,* 2004b).

Elles constituent un outil intéressant par leur caractère génératif. Elles permettent une description hiérarchisée des produits faisant correspondre les décompositions aux différentes règles de réécriture. Toutefois l'emploi des grammaires de webs pour la modélisation de famille de produits a des limites. Son utilisation pour la modélisation de familles de produits nécessite la connaissance de la famille en extension lorsqu'il existe un nombre important de variantes. Par ailleurs ce dernier aspect est source d'explosion combinatoire doublée de la difficulté de mise à jour des modèles de famille de produits.

Pour pallier ces difficultés, nous allons montrer que l'utilisation de la modélisation par les grammaires de graphes est une solution efficace.

## 4. Méthodologie de modélisation des familles de produits par les grammaires de graphes

L'expression *grammaire de graphe* se réfère généralement à une variété de méthodes pour la spécification (possibilités infinies) des ensembles de graphes (Jiao et Tseng, 1999). Typiquement, les nœuds représentent les objets ou concepts, les arêtes représentent les relations entre eux et l'attribut d'arête représente l'information non structurelle associée aux objets ou relations.

Les entités de famille de produits relatives aux éléments de grammaire de graphe (produits, produits de base, modules distinctifs) sont représentées par des graphes. Les modules sont modélisés comme des nœuds, les interconnexions (interfaces) entre ces modules comme des arêtes et les paramètres de modules comme attributs du nœud. Les combinaisons des modules ainsi que les conditions

relatives au moment où ces combinaisons doivent être mises en place sont modélisées respectivement comme opérations et conditions d'application des règles de production. En adoptant le concept de grammaire de graphe à attribut programmée (Programmed Attributed Graph Grammars, PAGG) (Bunke, 1982), la séquence pour l'exécution d'un ensemble de productions est exprimée par un diagramme de contrôle. La description statique d'une famille de produits est composée des notations de ces grammaires de graphe. Les variantes de la famille de produits peuvent être décrites par application des règles de production tributaires du diagramme de contrôle pour modifier le graphe initial, graphe représentant le produit de base (ou initial). Les graphes résultants sont les modèles des variantes désirées.

Tous ces graphes composent le langage de cette grammaire, qui décrit l'espace de conception de la famille de produits.

### 4.1. *Exemple industriel*

L'exemple illustrant ce travail est un produit d'une entreprise de 110 salariés, fabriquant *des meubles de cuisines et de salles de bain et* ayant des types de clients variés (ménages, entreprise etc…). La cuisine intégrée est par excellence un produit à forte diversité et à caractère modulaire. Ce type de produit est constitué d'éléments organisés en caissons, portes, quincailleries, poignées et boutons remplissant différents rôles : éléments sous évier, éléments de rangement, éléments d'angle. La diversité est obtenue par le choix des formes (type de moulure, type de contournement...), des finitions (placages, coloris, types de vernis ...), des quincailleries (poignées, boutons, piètement, ...), des dimensions multiples, etc…

Pour le meuble qui sera analysé par la suite il y a un choix de 11 façades et de 5 piètements. L'entreprise offre trois types de quincaillerie, dont le choix est fixé soit par des considérations esthétiques liées au choix de la façade, soit par des considérations techniques.

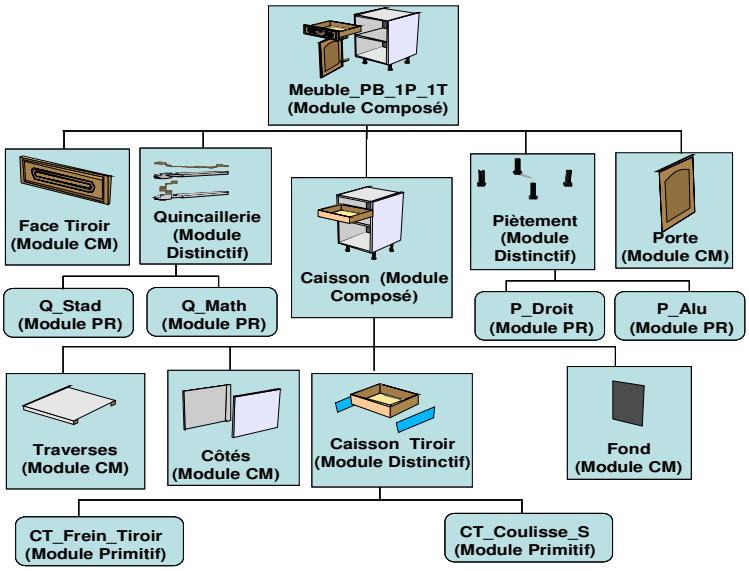



**Figure 2.** *Représentation du produit Meuble Partie basse Une Porte Un Tiroir*

Nous nous intéressons plus particulièrement à un meuble de rangement bas (voir figure 2). Il a l'avantage d'être un élément de cuisine adaptable à la salle de bain. Il constitue la famille de produits Meuble Partie Basse Une Porte Un tiroir (MPB_1P_1T) représentée à la figure 1.

A partir des définitions de la section 2.3, la SGP de la famille de meubles est représentée sur la figure 2. Les sous-systèmes communs ou composants à partir desquels les variantes d'un produit peuvent être obtenues seront dénommés *produit de base*. Dans la pratique, le produit de base peut prendre plusieurs formes. Il peut être un produit avec les fonctionnalités de base, un produit semi-fini ou un produit standard qui remplit les exigences de la majorité des clients. Du point de vue technique (industriel), le produit de base matérialise la commonalité dans une famille de produits. Dans notre exemple (figure 4), le produit de base est constitué des modules communs fond (FOND), côtés (COTE), traverse haute (TRA_H) et traverse basse (TRA_B) ainsi que des interactions des autres modules avec le caisson, ici le module porte (PORT). A partir du produit présenté ci-dessus, il est possible de générer 27 variantes, les modules CF_FT et CF_FS pouvant apparaître séparément ou simultanément.

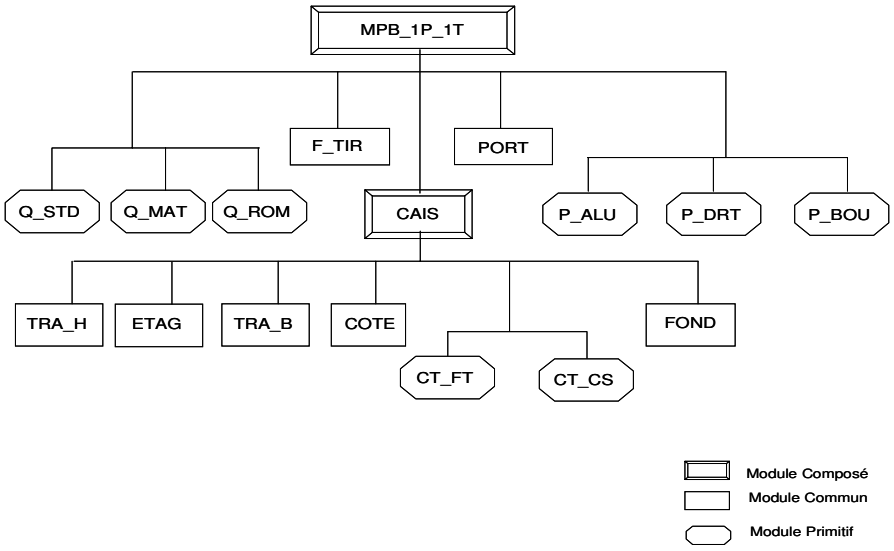



**Figure 3.** Structure générique de la famille de produits MPB_1P_1T

### 4.2. *Formalisation du graphe*

Une grammaire de graphe à attribut programmée est définie par six éléments :

- les étiquettes de sommet
- les étiquettes d'arc
- les attributs de sommet
- le graphe initial
- les productions
- le diagramme de contrôle.

Par conséquent, la grammaire de graphe pour un module M est un 6-uplet défini comme suit :

$$G(M) \cong (V, W, A, G^i, P, C) \qquad [3]$$

Où

$V$ est l'ensemble constitué des sommets étiquetés (i.e. des noms) de tous les modules.

$W$ représente l'ensemble constitué des arcs étiquetés qui indiquent les liaisons entre les modules de $M$ .

$A$ est l'ensemble constitué des attributs de sommet représentant les paramètres des modules fils (ils permettent l'instanciation des modules).

$G^i$ caractérise le graphe initial représentant le produit de base de M.

$P$ est l'ensemble comprenant toutes les règles de productions définies pour les manipulations des modules distincts afin de générer les variantes de M.

$C$ matérialise le diagramme de contrôle sur $P$ spécifiant l'ordre par lequel les productions sont appliquées afin que les variantes de $M$ puissent être obtenues.

En l'appliquant au caisson qui est un module composé, l'équation [3] s'instancie comme suit (les symboles sont tous référencés dans l'annexe):

$$G(M_{CAIS}) \cong (V_{CAIS}, W_{CAIS}, A_{CAIS}, P_{CAIS}, G^{i}_{CAIS}, C_{CAIS}) \quad [4]$$

Avec

$V_{CAIS} = \{M_{CT\_FT}, M_{CT\_CS}, M_{F\_TIR}, M_{COTE}, M_{TRA\_H}, M_{TRA\_B}, M_{PORT}, M_{FOND}, M_{S\_GLI}\}$ représente l'ensemble de tous les modules apparaissant dans les graphes encapsulés du produit de base et des modules distinctifs.

$W_{CAIS} = \{gli, fix, enc, piv\}$ est l'ensemble des interfaces des sommets étiquetés qui apparaissent dans les graphes du produit de base et des modules distinctifs.

$A_{CAIS}$ est l'ensemble des attributs des sommets représentant les paramètres des modules fils qui ne nous sont pas utiles dans ce cas de figure compte tenu du fait que la génération de la variété ne nécessite pas cette spécification.

$P_{CAIS} = \{p_{CT\_FT}, p_{CT\_CS}\}$ est l'ensemble des productions des modules distincts permettant de générer les variantes explicitées à l'équation [12] du paragraphe 4.5.

$G^{i}_{CAIS} = G(M^{0}_{CAIS})$ est le graphe initial représentant le produit de base présenté à la figure 4-1.

$C_{CAIS}$ : le diagramme de contrôle du caisson spécifiant l'ordre dans lequel les productions sont appliquées est représenté à la figure 9.

Le tableau 1 (en annexe) synthétise les dénominations des modules et des éléments.

Le tableau 2 (en annexe) donne la nomenclature des liaisons.

### 4.3. *Modélisation des interfaces et de la structure interne : graphe encapsulé*

Pour caractériser les interfaces externes et les structures internes des modules, on utilise la notion de graphe encapsulé. Le concept de graphe encapsulé a été introduit par (Engels et Schürr, 1995). Il est utilisé pour décrire différents niveaux d'abstraction de systèmes hiérarchiques. Nous utilisons cette notion pour la modélisation des modules.

Le graphe encapsulé (*graphe_e*) $g$ du module $M$ est spécifié par un 7-uplet défini comme suit :

$$g \cong (S_c, S, A_c, A, S_{cp}, S_{pr}, S_{cm}) \qquad [5]$$

Où

$S_c(g)$ est l'ensemble constitué de sommets connus de $g$ qui correspond à tous les modules contenus ou reliés au produit, ainsi les modules contenus dans le produit sont appelés modules fils et ceux qui sont reliés au produit appelés modules de contexte.

$S(g)$ est l'ensemble comprenant tous les sommets appartenant à $g$ et correspondant à tous les modules fils de $M$.

$A_c(g)$ est l'ensemble comprenant tous les arcs connus de $g$.

$A(g)$ est l'ensemble comprenant tous les arcs qui appartiennent à $g$.

$S_{cp}(g)$ représente l'ensemble de tous les sommets composés correspondant aux modules fils composés de $M$.

$S_{pr}(g)$ est l'ensemble de tous les sommets primitifs qui correspondent aux modules fils primitifs de $M$.

$S_{cm}$ est l'ensemble de tous les sommets communs correspondant aux modules fils communs de $M$.

Tous ces ensembles sont formalisés par les relations [5.1] et [5.2] suivantes :

$$\begin{cases} S_c(g) = S_c \ ; S_c \neq \varnothing \\ S(g) = S \ ; S \subseteq S_c, S \neq \varnothing \\ A_c(g) = A_c \subseteq S_c \times S_c \\ A(g) = A \subseteq S \times S \ ; A \subseteq A_c \\ S_{cp}(g) = S_{cp} \subseteq S \\ S_{pr}(g) = S_{pr} \subseteq S \\ S_{cm}(g) = S_{cm} \subseteq S \\ S(g) = S_{cp}(g) \cup S_{pr}(g) \cup S_{cm}(g) \end{cases} \qquad [5.1]$$

$$\begin{cases} S_{ct}(g) = S(g) \big/ S_c(g) \\ A_{ct}(g) = A(g) \big/ A_c(g) \\ G(M) \end{cases} \qquad [5.2]$$

Où

$S_{ct}(g)$ représente l'ensemble composé de tous les sommets de contexte dans g, c'est le complément de $S_c(g)$ par rapport à $S(g)$.

$A_{ct}(g)$ représente l'ensemble de tous les arcs de contexte dans g, c'est le complément de $A_c(g)$ par rapport à $A(g)$.

$G(M)$ définit le graphe encapsulé (graphe_e) du module $M$.

Les notations de l'équation [5.2] seront utilisées par la suite, en particulier au tableau 3 de l'annexe.

A partir du graphe encapsulé on peut définir toutes les catégories de modules dans une famille de produits comme le montre les équations [5.3] à [5.6]. En effet,

$$\text{Si} \quad S_{cp}(G(M)) \cup S_{pr}(G(M)) = \varnothing, \ M \text{ est un module commun} \qquad [5.3]$$

Si $\left|S(G(M))\right| = 1$, M est un module primitif [5.4]

Si $\left|S(G(M))\right| > 1$, M est un module composé [5.5]

Si $\left|S(G(M))\right| > 1$ et $S_c(G(M)) = \varnothing$, M est un produit final [5.6]

Dans le cas du caisson, la première étape consiste à définir les graphes encapsulés du produit de base et de chaque module distinctif. La figure 6-1 présente le graphe encapsulé du produit de base, on peut remarquer la présence des modules communs fond (FOND), côté (COTE), traverse haute (TRA_H) et traverse basse (TRA_B) ainsi que les interactions des autres modules avec le caisson. Les étiquettes d'arc représentent les liaisons (liaison pivot : piv, liaison fixe : fix).

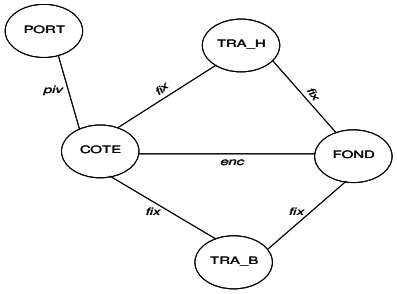



**Figure 4-1.** *Graphe encapsulé du produit de base "caisson"*

La figure 4-2 représente le graphe encapsulé du module distinctif caisson tiroir à frein tiroir (CT_FT), les interfaces (liaison fixe (fix) et liaison glissière (gli)) sont également matérialisées.

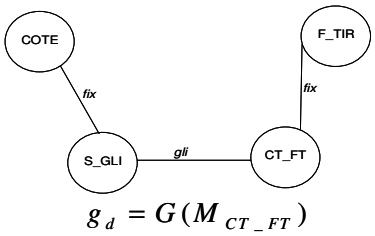



**Figure 4-2.** *Graphe encapsulé du module distinctif "Caisson Tiroir à Frein Tiroir"*

De même la figure 4-3 représente le graphe encapsulé du module distinctif caisson tiroir à coulisse standard (CT_CS) ainsi que ses interfaces.

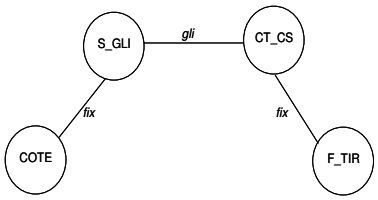



**Figure 4-3.** *Graphe encapsulé du module distinctif "Caisson Tiroir à Coulisse Standard"*

Le tableau 1 (en annexe) synthétise les dénominations des modules et des éléments.

Le tableau 2 (en annexe) donne la nomenclature des liaisons.

Le tableau 3 (en annexe) résume la définition textuelle de ces graphes encapsulés définis aux équations [5], [5.1] et [5.2].

## 4.4. *Génération de la variété*

Basée sur l'architecture modulaire, la variété des produits s'obtient par des combinaisons variées de modules (Ulrich, 1995). Elle résulte aussi des modifications topologiques ou de la variation des attributs du sommet. La génération de la variété se définit comme une spécification des transformations de l'état des graphes associés. La variété résulte de quatre opérations (Ulrich, 1991 ; Du *et al.*, 2002): l'ajout, l'élimination, l'échange et l'ajustement de module.

### *4.4.1. Opération d'ajout de module*

Ajouter un module $M$ (noté *ajou*) à un produit de base $PB^0$, est équivalent à lier le *graphe_e* $G(M)$ au graphe hôte $g = g(PB^0)$ aussi longtemps qu'il y aura des sommets de contexte de $g(M)$ dans $G(PB^0)$. Elle est matérialisée par le 3-uplet $(g_g, g_d, T_i)$ défini comme suit :

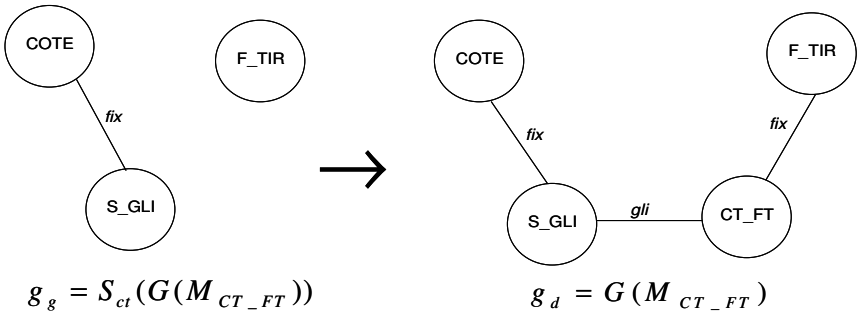
$$\text{ajou}(G^{-1}(G(M))) = \text{ajou}(M) \qquad [6]$$

Où

$G^{-1}(g_d)$ est le module représenté par le *graphe_e* de $g_d$. $g_g$ est un sous graphe de $g$ et est composé des sommets de contexte de $G(M)$, ce qui signifie que le produit de base met à disposition l'interface avec le module $M$ qui doit être ajouté. $T_i$ est une fonction de transformation intégrée spécifiant que tous les arcs connectées aux sommets dans le $g_g$ devront être connectés aux sommets correspondant dans le $g_d$. Les équations [6.1] et [6.2] montrent le formalisme général et une instanciation (ajout du module $M_{CT\_FT}$) au produit de base.

$$\begin{cases} g_g = S_{ct}(G(M)) \\ g_d = G(M) \\ T = (g_g, S_{ct}(g_d)) \end{cases} \quad [6.1]$$

$$\begin{cases} ajou(M_{CT\_FT}) = (g_g, g_d, T_i) \\ g_g = S_{ct}(G(M_{CT\_FT})) \\ g_d = G(M_{CT\_FT}) \\ T_i = (g_g, S_{ct}(g_d)) \end{cases} \quad [6.2]$$




$g_g = S_{ct}(G(M_{CT\_FT}))$ $g_d = G(M_{CT\_FT})$

**Figure 5.** *Opération d'ajout du module "Caisson Tiroir à Frein Tiroir"*

*4.4.2. Opération d'élimination de module*

Une variante différente est obtenue par le retrait d'un module possédant un certain nombre de fonctions redondantes dont n'a pas besoin le client. Ce module est retiré dans le but d'avoir un produit simplifié et plus économique.

Eliminer un module M (noté elim) d'un produit de base $PB^0$ équivaut à éliminer le *graphe_e* $G(M)$ du graphe hôte $g = G(PB^0)$ mais à garder les nœuds (sommets) de contexte de $G(M)$ dans le graphe résultat, il consiste à faire disparaître une variante de la famille. Il est matérialisé par le 2-uplet $(g_g, g_d)$ défini comme suit :

$$e\lim(G^{-1}(G(M))) = e\lim(M) \quad [7]$$

Où

$g_d$ est composé de sommets de contexte de $G(M)$ et est un sous graphe du graphe résultant.

$$\begin{cases} g_g = G(M) \\ g_d = S_{ct}(G(M)) \end{cases} \quad [7.1]$$

En lisant la figure 9 dans le sens inverse, on identifie de manière simple que $g_g = G(M_{CT\_FT})$ et $g_d = S_{ct}(G(M_{CT\_FT}))$.

*4.4.3. Opération d'échange de module*

Plusieurs modules ayant des performances différentes peuvent réaliser la même fonction. La variété est alors due au niveau de performance requis. Cela nécessite

bien sûr que les modules possèdent la même interface. On peut aussi remarquer que l'échange des modules peut être considéré comme une combinaison d'ajout et d'élimination.

Echanger (noté *echa*) consiste à remplacer un module $M^0$ dans un produit de base $PB^0$ par un module $M$ qui partage avec $M^0$ les interfaces identiques sur les autres parties du produit. Il s'agit d'une substitution du graphe encapsulé *graphe_e* $G(M)$ par $G(M^0)$ dans le graphe hôte $g = G(PB^0)$ aussi longtemps que $G(M)$ et $G(M^0)$ posséderont les mêmes sommets de contexte. C'est un 3-uplet $(g_g, g_d, T_i)$ défini comme suit :

$$echa(G^{-1}(G(M))) = echa(M) \quad [8]$$

Avec

$$\begin{cases} g_g = G(M^0) \\ g_d = G(M) \\ S_{ct}(g_d) = S_{ct}(g_g) \\ T_i = (S_{ct}(g_g), S_{ct}(g_d)) \end{cases} \quad [8.1] \quad \begin{cases} echa(M_{CT\_FT}) = (g_g, g_d, T) \\ g_g = G(M_{CT\_CS}) \\ g_d = G(M_{CT\_FT}) \\ T = (S_{ct}(g_g), S_{ct}(g_d)) \end{cases} \quad [8.2]$$

Les équations [8.1] et [8.2] montrent le formalisme général et son instanciation à l'échange entre les modules $M_{CT\_FT}$ et . La fonction de transformation intégrée $T_i$ matérialise le fait que tous les arcs connectés aux sommets de contexte de $g_g$ seront reliés aux sommets correspondant dans $g_d$. $M_{CT\_FT}$ et $M_{CT\_CS}$ sont les variantes du module $M_{C\_TIR}$.

### *4.4.4. Opération d'ajustement de module*

Dans certains cas pour garder la même structure pour le produit, il est nécessaire soit d'améliorer soit de dégrader les performances d'un module. Ce qui se traduit par une modification des options liées à ce module, cette opération sera appelée ajustement.

Ajuster un module (noté *ajus*) consiste à redimensionner les paramètres du module $M$. C'est équivalent à changer les attributs du sommet $S(M)$ dans le graphe hôte $G(M)$. $S(M)$ est un sommet associé à $M$ alors que la topologie du graphe demeure inchangée.

Une opération de redimensionnement est matérialisée par un 3-uplet $(g_g, g_d, f)$ défini comme suit :

$$ajus(G^{-1}(G(M))) = ajus(M) \quad [9]$$

Où

$f$ est la fonction de transfert attribut du sommet $S(M)$.

$$\begin{cases} g_g = g_d = G(M) \\ s_g = s_d = S(M) \in S(G(M)) \\ \alpha(s_g) \neq \alpha(s_d) \end{cases} \quad [9.1] \quad \begin{cases} ajus(M_{PORT}) = (g_g, g_d, f) \\ g_g = g_d = G(M_{PORT}) \\ \alpha(S(M_{PORT})) = f(\alpha(S(Parent))) \end{cases} \quad [9.2]$$

A travers cette opération, l'attribut de $M_{PORT}$ passe de $A_{PORT} = \alpha(S(M_{PORT}))$ à $A'_{PORT} = \alpha(S(Parent))$. Ici, l'attribut $A_{PORT}$ est l'ajustement du module porte à l'épaisseur 2 cm tandis que l'attribut $A'_{PORT}$ est l'ajustement du module porte à l'épaisseur 3 cm.

### 4.5. *Génération des variantes et des productions*

Après avoir défini les quatre opérations sur les graphes permettant de générer la variété, il reste à spécifier les conditions sous lesquelles ces opérations seront exécutées. Pour ce faire, (Du *et al.,* 2002) introduisent le prédicat d'applicabilité $\pi$ (qui peut être VRAI ou FAUX) comme conditions d'applications à satisfaire.

Une production est définie par un 2-uplet comme suit :

$$p \cong (O, \pi) \quad [10]$$

Où

$$\begin{cases} \forall O \in \{ajou, elim, echa, ajus\} \\ \forall \pi \in \{VRAI, FAUX\} \end{cases} \quad [10.1]$$

O représente ici l'opération à effectuer. Toutefois, si des opérations n'appartenant à l'ensemble {*ajou, elim, echa, ajus*} sont exigées, elles pourront être rajoutées à l'ensemble. $\pi$ est le prédicat d'applicabilité, c'est une fonction logique des conditions d'applications.

#### *4.5.1. Définition et mise en place des productions*

L'exécution normale des productions nécessite de détailler les spécifications de la famille de produits MPB_1P_1T en termes de fonctions (communes, optionnelles, contraintes de sélection) et d'options. Nous illustrerons seulement l'ajout de modules.

Pour le caisson, le système glissière et la face tiroir sont des modules communs. On peut ensuite ajouter un caisson tiroir à frein tiroir (CT_FT) et / ou un caisson tiroir à coulisse standard (CT_CS). L'ajout de ces modules est déterminé par le client.

Le client peut exiger aussi bien un module face tiroir muni d'un système d'ouverture à poignée (module CT_CS) qu'une face tiroir ayant un système d'ouverture à bouton nécessitant le module CT_FT, certains clients par contre exigent les deux, CT_FT + CT_CS. La fonction optionnelle, exigence des clients, sera matérialisée par $F_{SO\_FT}$ et définie par la relation [11] comme suit :

$$\begin{cases} F_{SO\_FT} : \text{système d'ouverture face tiroir} \\ \text{option}_{SO\_FT} \in \{CT\_FT, CT\_CS, CT\_FT \text{ et } CT\_CS\} \end{cases} \quad [11]$$

La définition des productions est caractérisée par les différentes combinaisons possibles des modules distinctifs suivant un certain nombre de contraintes. Deux productions possibles sont présentées aux figures 8-1 et 8-2. Les graphes qui sont à gauche et à droite du symbole $\rightarrow$ sont respectivement la partie gauche du graphe (PGG) et la partie droite du graphe (PDG) des productions associées.

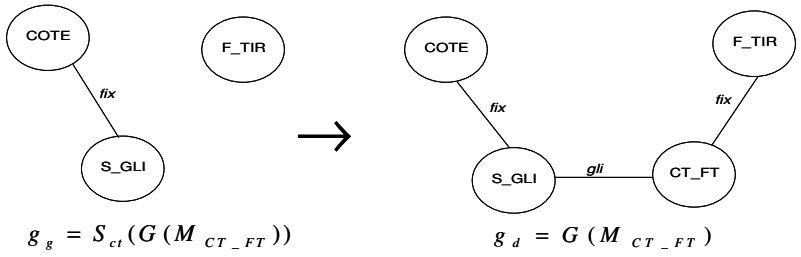



**Figure 6-1**. *Production* $p_{CT\_FT}$ *représentant l'ajout d'un module*

La production $p_{CT\_FT}$, présentée à la figure 8-1 permet d'ajouter le caisson tiroir à frein tiroir CT_FT (sommet CT_FT) au produit de base. Ceci n'est possible que dans le cas où le système d'ouverture de la face tiroir du caisson est muni d'un système d'ouverture à bouton, c'est-à-dire CT_FT ou CT_FT + CT_CS. On remarque que la PGG de la production est constituée des sommets de contextes (voir tableau 3) du graphe encapsulé du caisson tiroir à frein tiroir tandis que la PDG représente le graphe encapsulé du même caisson tiroir à frein tiroir.

La production $p_{CT\_CS}$, comme le montre la figure 8-2 est celle permettant d'ajouter le module CT_CS (sommet CT_CS) au produit de base au cas où le système d'ouverture de la face tiroir du caisson est muni d'une poignée, soit CT_CS ou CT_CS + CT_FT. Naturellement, la PGG de la production est constituée des sommets de contexte du graphe encapsulé du caisson tiroir à coulisse standard et la PDG, celle du graphe encapsulé du caisson tiroir à coulisse standard.

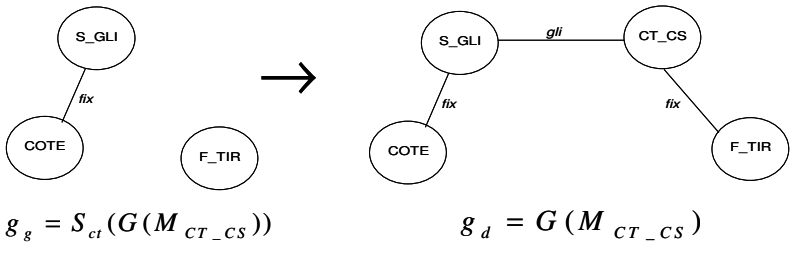



**Figure 6-2**. *Production* $p_{CT\_CS}$

Pour chaque production $p_{CT\_FT}$ *ou* $p_{CT\_CS}$, le prédicat d'application $\pi$ est une fonction logique de $F_{SO\_FT}$. Si $F_{SO\_FT}$ prend l'une des valeurs CT_FT (pour $p_{CT\_FT}$) ou CT_CS (pour $p_{CT\_CS}$), alors $\pi$ est vrai. Autrement $\pi$ est faux. Les spécifications textuelles de ces productions sont données par les équations [12].

$$\begin{cases} p_{CT\_FT} = (\text{ajou}((M_{CT\_FT}), \pi = (\alpha(F_{SO\_FT}) \in \{\{CT\_FT\}, \{CT\_FT + CT\_CS\}\}))) \\ p_{CT\_CS} = (\text{ajou}((M_{CT\_CS}), \pi = (\alpha(F_{SO\_FT}) \in \{\{CT\_CS\}, \{CT\_CS + CT\_FT\}\}))) \end{cases} \quad [12]$$

*4.5.2. Définition des diagrammes de contrôle*

La définition des diagrammes de contrôle a pour objectif de spécifier l'ordre d'application des productions. La figure 7 représente le diagramme de contrôle pour la famille de produit caisson. En dehors du sommet initial I et du sommet final F, tous les sommets sont étiquetés par des productions. Dans le diagramme, deux productions coexistent, donc les deux variantes du caisson réalisables seront $p_{CT\_FT}$ *et* $p_{CT\_CS}$.

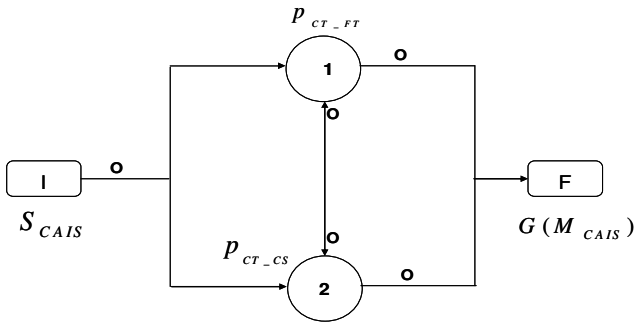



**Figure 7.** *Diagramme de contrôle*

Le diagramme a deux sommets hors I et F qui sont les productions $p_{CT\_FT}$ et $p_{CT\_CS}$.

Les variantes du produit sont obtenues en parcourant le diagramme. Chaque chemin indique un processus de génération possible d'une variante, par exemple, $I \to p_{CT\_FT} \to F$, $I \to p_{CT\_CS} \to F$, $I \to p_{CT\_FT} \to p_{CT\_CS} \to F$, $I \to p_{CT\_CS} \to p_{CT\_FT} \to F$.

Pour chaque chemin, la valeur initiale de l'attribut *Atteindre* est fixée à 1. Le caractère bidirectionnel de l'arc entre les sommets $p_{CT\_FT}$ *et* $p_{CT\_CS}$ implique

que le produit de base peut être modifié par l'ajout de l'un ou l'autre module. Dit autrement, le choix d'une direction permettra de considérer le module CT_FT en premier et le module CT_CS en second, tandis que le choix de la direction contraire permettra de considérer le module CT_CS d'abord et le module CT_FT ensuite. Dans cet exemple, il n'y aura pas d'ordre pour réaliser l'ajout.

*4.5.3. Dérivation de variante*

La structure hiérarchique des produits est considérée comme une collection de modules composés organisés à travers plusieurs niveaux d'abstraction. La dérivation de variante de produit sera un processus récursif, générateur de tous les modules composés. Ce processus progresse de manière récursive pour les autres modules composés, ceci jusqu'à l'atteinte du niveau le plus bas des modules composés. Une fois le niveau le plus bas atteint, les modules fils des modules composés ne sont autres que les modules primitifs ou communs, par conséquent la détermination des variantes de ces modules composés peut se faire localement. Le processus de génération de ces variantes est décrit dans (Du et *al.*, 2002), l'application faite dans le cas de la famille de meubles est décrite dans (Mtopi, 2006).

## 5. Discussion et conclusion

Nous venons de présenter dans ce texte l'analyse d'un outil mathématique appliqué à la génération de la diversité dans les familles de produits : les grammaires de graphes à attributs programmées. Nous l'avons replacé dans le contexte plus général des modélisations par grammaire (grammaires formelles et grammaires de web) afin d'en faire apparaître les intérêts principaux.

Les grammaires de graphe par attributs programmés représentent un outil intéressant pour la modélisation des familles de produits surtout pour leur caractère génératif. Elles permettent une description hiérarchisée des produits faisant correspondre les décompositions aux différentes règles de réécriture.

Cet outil a le double avantage de s'appuyer sur une formulation mathématique et des représentations graphiques. Son choix pour la modélisation d'une famille de produits est justifié car cet outil permet de mettre en évidence les liaisons complexes entre les modules et composants de la famille à travers l'identification de ces liaisons (arcs étiquetés) ainsi que la manipulation de la génération des variantes.

L'approche par les grammaires de graphe a pour point de départ la structure générique de produit s'appuyant sur la notion fondamentale de modules (communs et distinctifs). Dans un premier temps, les produits, produits de base et les modules sont représentés par des graphes (attributs et encapsulés). L'architecture modulaire des produits conduit à la manipulation de productions, basées sur quatre principales opérations (ajout, élimination, échange et d'ajustement). Ces opérations permettent d'une part de générer la variété et d'autre part d'agir sur la dérivation des produits, ce qui pourrait conduire à des variantes non identifiées au départ.

Les différentes combinaisons des modules sont modélisées par l'application des productions qui sont synchronisées non seulement à travers la définition des

conditions, mais aussi à travers le contrôle de l'ordre d'exécution des dites productions, piloté par le diagramme de contrôle.

Toutefois, l'existence d'un ordre imposé de lecture des règles dans le diagramme de contrôle complique tout traitement puisque d'autres ordres d'accès aux données peuvent être requis. En général du point de vue strictement formel, les structures générées sont correctes, mais leur sens demanderait parfois à être validé par un expert dans le métier.

L'utilisation des règles de génération des variantes devrait permettre de coder un savoir sur les produits, de le partager et de le réutiliser, et ainsi de développer une capitalisation des connaissances réutilisables de façon appropriés par les concepteurs.

Des travaux sur un produit similaire ont été présentés dans (Aldanondo et al, 2007). La variété est générée par des mécanismes de résolution et de propagation de contraintes. Contrairement à notre produit modélisé par graphes, le modèle produit n'est ni structuré, ni hiérarchisé. Le produit est vu comme un ensemble de composants représenté par une nomenclature en râteau. La configuration de produit a ensuite été couplée à la configuration de sa gamme de production. Selon les auteurs les approches par contraintes permettraient de traiter des problèmes de grande taille et complexité élevée.

L'absence actuelle de plateforme logicielle supportant un modèle à base de graphes ne permet pas de pouvoir comparer l'approche proposée avec les résultats obtenus par un configurateur. Les difficultés de la modélisation et de la génération de la diversité par les graphes résident dans la définition complète du diagramme de contrôle et sa modification en cas d'introduction d'une nouvelle variante. Cette méthode, comme la plupart des autres, présuppose une identification préalable des modules communs/distinctifs et la définition de l'architecture de la famille.

## 6. Annexe

Le *meuble partie basse une porte un tiroir* a pour principaux modules : la porte, la face tiroir,la quincaillerie (pouvant être de type Mathilde, Romane, standard, etc.), le piètement ( piètement boule, aluminium, droit, etc.), le caisson, lui-même composé : des 2 côtés, du caisson tiroir (caisson tiroir à coulisse standard et / ou caisson tiroir à frein tiroir), du fond, de traverses (traverse haute, traverse basse, étagère), des deux systèmes glissières permettant le mouvement du tiroir.

Pour des besoins de codification et afin de mieux représenter la structure générique du produit (SGP), les graphes encapsulés, les règles de production et les diagrammes de contrôle, nous proposons la nomenclature suivante représentée au tableau 1.

| Dénomination du module | Nomenclature | Dénomination du module | Nomenclature |
|---|---|---|---|
| Meuble Partie Basse Une porte Un Tiroir | MPB_1P_1T | Piètement Boule | P_BOU |
| Porte | PORT | Caisson | CAIS |
| Face Tiroir | F_TIR | Système Glissière | S_GLI |
| Etagère | ETAG | Caisson Tiroir à Frein Tiroir | CT_FT |
| Quincaillerie Standard | Q_STD | Caisson Tiroir à Coulisse Standard | CT_CS |
| Quincaillerie Mathilde | Q_MAT | Cotés | COTE |
| Quincaillerie Romane | Q_ROM | Fond | FOND |
| Piètement Droit | P_DRT | Traverse Haute | TRA_H |
| Piètement Aluminium | P_ALU | Traverse Basse | TRA_B |

**Tableaux 1.** Nomenclature des modules de la famille de produits MPB_1P_1T

Les relations entre les modules sont les interfaces entre les modules et sont caractérisées par les types de liaisons existantes. Nous avons identifié les types de liaisons suivantes entre les modules.

| Dénomination du type de liaison | Nomenclature |
|---|---|
| Rotation directe = liaison pivot | piv |
| Solidarisation par tourillon = liaison fixe | fix |
| Solidarisation par agrafe = liaison fixe | fix |
| Solidarisation par vis = liaison fixe | fix |
| Rainure = liaison par encastrement | enc |
| Solidarisation par emboitement = liaison par encastrement | enc |
| Glissière = liaison glissière | gli |

**Tableau 2.** *Nomenclature des différentes liaisons*

| | $G(M^{0}_{CAIS})$ | $G(M_{CT\_FT})$ | $G(M_{CT\_CS})$ |
|---|---|---|---|
| $S_c$ | {(PORT, FOND, COTE, TRA_H, TRA_B} | {CT_FT, COTE, F_TIR, S_GLI} | {CT_CS, COTE, F_TIR, S_GLI} |
| $S$ | {FOND, COTE, TRA_H, TRA_B} | {CT_FT} | {CT_CS} |
| $A_c$ | {(PORT, COTE), (FOND, COTE), (FOND, TRA_H), (FOND, TRA_B), (TRA_H, COTE) | {(COTE, S_GLI), (S_GLI, CT_FT), (CT_FT, F_TIR)} | {(COTE, S_GLI), (S_GLI, CT_CS), (CT_CS, F_TIR)} |
| $A$ | ∅ | ∅ | ∅ |
| $S_{cp}$ | ∅ | ∅ | ∅ |
| $S_{pr}$ | ∅ | {CT_FT} | {CT_CS} |
| $S_{cm}$ | ∅ | ∅ | ∅ |
| $S_{ct}$ | {PORT} | {COTE, F_TIR, S_GLI} | {COTE, F_TIR, S_GLI} |
| $A_{ct}$ | {(PORT, COTE), (FOND, COTE), (FOND, TRA_H), (FOND, TRA_B), (TRA_H, COTE) | {(COTE, S_GLI), (S_GLI, CT_FT), (CT_FT, F_TIR)} | {(COTE, S_GLI), (S_GLI, CT_CS), (CT_CS, F_TIR)} |

**Tableaux 3.** *Définition des graphes encapsulés de la famille de produits caisson*

## 7. Bibliographie


Agard B., « Conception et fabrication des familles de produits : Etat de l'art », RS *Journal Européen des Systèmes Automatisés - JESA*, vol. 38 (1-2), 2004, p. 59-84.

Agard B., Tollenaere M., « Conception d'assemblage pour la customisation de masse », *Mécanique et industrie*, vol. 3, 2002, p. 113-119.

Agard B., Tollenaere M., « Méthodologie de conception des familles de produits », *Journal Européen des Systèmes Automatisés - JESA*, vol. 37, n°3, 2003, p. 755-777.

Aldanondo M., Vareilles E., Hadj-Hamou K., Moynard G., « Une approche pour la configuration cohérente de produit et de mode opératoire pour la simulation », *Proceedings de la 6ème Conférence Francophone de Modélisation Optimisation et SIMulation – MOSIM'06*, Rabat, avril 2006, Maroc.

Aldanondo M., Vareilles E.,Lahmar Y., Baron C., Moynard G., « Une approche pour la configuration cohérente de produit et de gamme de production», RS *Journal Européen des Systèmes Automatisés - JESA*, vol. 41, 5/ 2007.

Baldwin C., Clark, K., "Managing in the Age of Modularity", Harvard Business Review, Vol 75, N° 5, pp. 84-93, September-October 1997

Bourjault A., Contribution à une approche méthodologique de l'assemblage automatisé : élaboration automatique des séquences opératoire, Thèse de doctorat d'état, Université de Franche-Comté, 1984.

Bunke H., « Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation », *IEEE Transaction on Pattern Analysis and Machine Intelligence - PAMI*, vol. 4, n°6, 1982, p. 574-582.

Chen, K.M., Liu, R.J.. « Interface strategies in modular product innovation », *Technovation*, vol. 25, 2005, p. 771–782

Danloy J., Petit F., Leroy A., De Lit P., Rekiek B., « A pragmatic approach for precedence graph generation », *Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning ISATP'99*, Porto, july 1999, Portugal, p. 387-392.

De Lit P., A comprehensive and integrated approach for the design of product family and its assembly system, Thèse de doctorat, Université libre de Bruxelles, Belgique, 2001.

Djemel N., Contribution à la conception des systèmes flexibles d'assemblage dans le cas multi - produits, Thèse de doctorat, Université de Franche-Comté, 1994.

Du X., Jiao J., Tseng M., M., «Graph Grammar Based Product Family Modeling», *Concurrent Engineering: Research and Applications - CERA*, vol. 10, n°2, 2002, p. 113-128.

Dufrène L., Contribution à une méthodologie de conception des systèmes d'assemblage pour familles de produits, Thèse de doctorat, Université de Franche-Comté, 1991.

Engels G., Schürr A., « Encapsulated Hierarchical Graphs, Graph Types, and Meta Type », *Electronic notes in Theoretical Computer Science*, vol. 2, 1995. URL: http://www.elsevier.nl/located/volumes2.html.

Erens F., J., The Synthesis of Variety: Developing Product Families, PhD Thesis, Technische Universiteit Eindhoven, The Netherlands, 1996.

Erens F., J., Verhulst K., « Architectures for product families », *Computer in Industry*, vol. 33, 1997, p. 165-178.

Fan I., S., Liu C., K., « Product family and variants: Definition and models », *Int J., Flexible Automation and Intelligent Manufacturing 1999,* Tilburg, June 1998, The Netherland, Asharyeri, W. G. Sullivan, and M.M. Ahmed, editors.

Fujita K., "Product variety optimization under modular architecture", Computer-Aided Design, Vol 34, pp.953-965, 2002.

Henrioud J., M., Contribution à la conceptualisation de l'assemblage automatisé : nouvelle approche en vue de la détermination des processus d'assemblages, Thèse de doctorat d'état, Université de Franche-Comté, 1989.

Jiao J., Tseng M., M., Duffy V., Lin F., « An information Modeling Framework for Product Families to support Mass customization, *CIRP Annals*, vol. 48, n°1, 1999, p. 93-98.

Jiao J., Tseng M., M., Zou Y., « Generic bill of materials and Operations for hig-variety production management », *Concurrent Engineering: Research and Applications - CERA*, vol. 8, n°4, 2000, p. 297-322.

Jiao R., Huang G., Tseng M., "Concurrent Enterprising for Mass Customization", Concurrent Engineering: Research and Applications, Vol 12, N°2, pp. 83-88, 2004.

Jiao J., Simpson T., Siddique Z., « Product family design and platform-based product development: a state-of-the-art review », *Journal of intelligent manufacturing*, special issue on Product family design and platform-based product development, 2006.

José A., Tollenaere, "Modular and platform methods for product family design: literature analysis", Journal of Intelligent Manufacturing, 16, 2005, 371-390;

Laakko T., Mäntylä M., « Feature-based modeling of product of families », *In ASME International Computer in Engineering Conference*, vol. 1, 1994, p. 45-54.

Lamothe J., Hadj-Hamou K., Aldanondo M., « An optimization model for selecting a product family and designing its supply chain», *European Journal of Operational Research*, vol. 169, 2006, p. 1030-1047.

Männistö T., Peltonen H., Martio A., Sulonen R., « Modelling generic product structures in STEP », *Computers-Aided Design,* vol. 30, n°14, 1998, p. 1111-1118.

Männistö T., Sulonen R., « Evolution of schema and individuals of configurable products », *Advances in conceptual Design*, Lecture Notes in Computer Sciences, 1999.

Martin M., Ishii K., "Design for variety: developing standardized and modularised product platform architectures", Research in Engineering Design, Vol 13, pp. 213-235, 2002.

McKay A., Bloor M., S., De Pennington A., « A framework for product data», *IEEE Transactions on Knowledge and Data Engineering,* vol. 8, n°5, 1996, p. 825-838.

Meunier M., Eléments méthodologiques pour la conception de systèmes flexibles d'assemblage, Thèse de doctorat, Université de Franche-Comté, 1989.

Meyer H., M., « Revitalize your product lines through continuous platform renewal», *Research Technology Management*, vol. 40, n°2, 1997, p. 17-28.

Meyer H., M., Lopez L., « Technology Strategy in a software products company», *Journal of product Innovation Management*, vol. 12, n°4, 1995, p. 294-306.

Miller T., Liberatore M., « Seasonal exponential smoothing with damped trends, an application for production planning», *International Journal of Forecasting*, vol. 9, n°4, 1993, p. 509-515.

Mtopi F., B., E., Contribution à une méthodologie de conception modulaire : modélisation de la diversité dans les familles de produits, Thèse de doctorat, Université de Franche-Comté, France, 2006.

Mtopi F., B., E., Dulmet M., Bonjour E., «Different models to support concurrent design of product family », *Proceeding of 5th International Conference on Integrated Design and Manufacturing in Mechanical Engineering-IDMME'04,* Bath, April 5-7, United Kingdom, 2004a.

Mtopi F., B., E., Dulmet M., Bonjour E., «Product family in concurrent design », *Proceeding of 10th IFAC/IFORS/IMACS/IFIP symposium on Large Scale Systems: Theory and Applications – LSS 2004,* vol. 1, Osaka, July 26-28, Japan, 2004b.

Perrard C., Lutz P., Bourjault A., Henrioud J., M., « The MAYRAS software; An efficient tool for the rational design of assembly systems », *Proceeding of the 1993 International Conference on Assembly,* Adelaide , November 1993, Australia, p. 77-85.

Pfaltz J., L., Rosenfeld A., « Web grammars », *Proc. Lst Int. Joint Conf. Artificial Intelligence,* Washington, D. C., 1969 p. 609-919.

Rampersad H., K., *Integrated and Simultaneous Design for Robotic Assembly*, Chichester, United Kingdom, Wiley Series in Product development: Planning, Designing, Engineering. John Wiley and Sons, Inc., 1994.

Rosenfeld A., Milgram D., L., « Web Automata and Web Grammars », *Machine Intelligence,* vol. 7, Wiley, New York, 1972, p. 307-324.

Sako M., « Modularity and outsourcing: the nature of eco-evolution of product architecture and organization architecture in the global automotive industry », *11ème rencontre Internationale de Gerpisa,* Paris, juin 2003.

Salvolainen, T., Beeckmann D., Groumpos P., Jagdev H., « Positioning of modeling approaches, methods and tools », *Computers in Industry,* vol. 25, n°3, 1995, p. 255-262.

Siddique Z., Rosen D., W., « On the applicability of product variety design concepts to automotive platform commonality », *Proceedings of the 1998 ASME Design Engineering Technical Conferences,* Atlanta, 1999.

Simpson T.W., "A concept exploration method for product family design", Ph.D Thesis, Woodruff School of Mechanical Engineering, Georgia Institute of Technology, 1998.

Stadzisz P., C., Contribution à une méthodologie de conception intégrée des familles des produits pour l'assemblage, Thèse de doctorat, Université de Franche-Comté, France, 1997.

Ulrich K., « The Role of Product Architecture in the manufacturing Firm », *Research Policy,* vol. 24, n°3, 1995, p. 418-440.

Ulrich K., Eppinger S., "Product Design & Development", 2nd Ed. McGraw-Hill, New York, 2000.

Ulrich K., Tung K., « Fundamentals of product modularity », *In Issues in Design Manufacture Integration - ASME, vol*. 39, 1991.

Whitney D., E., « Computer-Aided design of flexible assembly systems – First Rapport », *Rapport CSDL – R- 1947,* 1995.

Zha X. F., Sriram R. D. " Platform-based product design and development: A knowledge-intensive Support Approach", In Knowledge-Based Systems" 19 (2006), 524-543;